**Parte II - Capitolo 7 - Stallo Dei Processi**

In un ambiente con multiprogrammazione più processi possono competere per ottenere un numero finito di risorse; se una risorsa non è correntemente disponibile, il processo richiedente passa allo stato d'attesa. Se le risorse richieste sono trattenute da altri processi, a loro volta nello stato d'attesa, il processo potrebbe non cambiare il suo stato. Situazioni di questo tipo sono chiamate di ***stallo*** (***deadlock***). Prima di adoperare una risorsa, un sistema deve richiederla e, dopo averla usata, deve rilasciarla. Un processo può richiedere tutte le risorse necessarie per eseguire un compito assegnatogli. Nelle ordinarie condizioni di funzionamento un processo può servirsi di una risorsa soltanto se rispetta la sequenza: richiesta, uso, rilascio. La richiesta e il rilascio devono avvenire per mezzo di chiamate di sistema. Una tabella di sistema registra lo stato di ogni risorsa e, se questa è assegnata, indica il processo relativo.

Una situazione di stallo si può verificare solo se si verificano contemporaneamente le seguenti condizioni: *mutua esclusione*, cioè una risorsa è utilizzabile da un solo processo per volta; *possesso e attesa*, cioè un processo in possesso di almeno una risorsa attende di acquisire risorse già in possesso da altri processi; *impossibilità di prelazione*, cioè una risorsa può essere rilasciata dal processo che la possiede solo volontariamente; *attesa circolare*, cioè deve esistere un insieme di processi tale che un processo attende una risorsa di quello successivo, che attende una risorsa da quello successivo e così via. Le situazioni di stallo si possono descrivere con maggior precisione avvalendosi di una rappresentazione detta ***grafo di assegnazione delle risorse***. Si tratta di un insieme di vertici, composto a sua volta da un sottoinsieme di processi e un sottoinsieme di risorse, e un insieme di archi. Un arco orientato da un processo a una risorsa è detto ***arco di richiesta***, mentre un arco orientato da una risorsa a un processo è detto ***arco di assegnazione***. Se il grafo non contiene cicli, nessun processo del sistema subisce uno stallo. Se ciascun tipo di risorsa ha esattamente un'istanza, allora l'esistenza di un ciclo implica la presenza di uno stallo. Se ogni tipo di risorsa ha più istanze, un ciclo non implica necessariamente uno stallo.

Il problema delle situazioni di stallo si può affrontare impiegando tre metodi: si può usare un protocollo per prevenire o evitare le situazioni di stallo, assicurando che il sistema non entri mai in stallo; si può permettere al sistema di entrare in stallo, individuarlo, ed eseguire il ripristino; si può ignorare del tutto il problema, fingendo che le situazioni di stallo non possano mai verificarsi nel sistema. Per ***prevenire le situazioni di stallo*** (*prevention*) bisogna assicurare che almeno una delle quattro condizioni necessarie perchè si verifichi una situazione di stallo non possa capitare. Per quanto riguarda la *mutua esclusione*, questa deve valere per le risorse non condivisibili. Poichè alcune risorse sono intrinsecamente non condivisibili, in generale non si possono prevenire le situazioni di stallo negando la condizione di mutua escluzione. Per assicurare che la condizione di *possesso e attesa* non si verifichi mai nel sistema, occorre garantire che un processo che richiede una risorsa non ne possegga altre. Si può usare un protocollo che ponga la condizione che ogni processo, prima di iniziare la propria esecuzione, richieda tutte le risorse che gli servono e che esse gli siano assegnate. Un protocollo alternativo è quello che permette a un processo di richiedere risorse solo se non ne possiede: un processo può richiedere risorse e adoperarle, ma prima di richiederne altre deve rilasciare tutte quelle che possiede. Entrambi i protocolli presentano due svantaggi: l'utilizzo delle risorse può risultare poco efficiente, poichè molte risorse possono essere assegnate, ma non utilizzate per molto tempo; si possono verificare situazioni di attesa indefinita se un processo richiede più risorse molto usate che risultano sempre assegnate a qualche altro processo. Per assicurare che l'*impossibilità di prelazione* non persista, si può impiegare un protocollo di questo tipo: se un processo che possiede una o più risorse ne richiede un'altra che non gli si può assegnare immediatamente (cioè il processo deve attendere), allora si esercita la prelazione su tutte le risorse attualmente in suo possesso. Si ha cioè un rilascio implicito di queste risorse, che si aggiungono alla lista delle risorse che il processo sta attendendo. Il processo viene nuovamente avviato quando può ottenere sia le vecchie risorse sia quella che sta richiedendo. Questo protocollo è adatto a risorse il cui stato si può salvare e recuperare facilmente in un secondo tempo, come i registri della CPU e lo spazio di memoria, mentre non si può in generale applicare a risorse come le stampanti e le unità a nastri. Un modo per assicurare che non si verifichi l'*attesa circolare* consiste nell'imporre un ordinamento totale all'insieme di tutti i tipi di risorse e un ordine crescente di numerazione per le risorse richieste da ciascun processo (vedere esempio del protocollo sul libro). Un algoritmo per ***evitare le situazioni di stallo*** (*avoidance*) richiede che ciascun processo dichiari il numero massimo delle risorse di ciascun tipo di cui necessita. Esso esamina dinamicamente lo stato di assegnazione delle risorse per garantire che non possa esistere una condizione di attesa circolare. Lo stato di assegnazione delle risorse è definito dal numero di risorse disponibili e assegnate e dalle richieste massime dei processi. Uno stato si dice *sicuro* se il sistema è in grado di assegnare risorse a ciascun processo in un certo ordine e impedire il verificarsi di uno stallo. Più formalmente, un sistema si trova in uno stato sicuro solo se esiste una *sequenza sicura*. Uno stato sicuro non è di stallo. Viceversa, uno stato di stallo è uno stato non sicuro. Non tutti gli stati non sicuri sono di stallo, in quanto uno stato non sicuro potrebbe condurre a uno stallo. Finchè lo stato rimane sicuro, il sistema operativo può evitare il verificarsi di stati non sicuri e quindi di stalli; in uno stato non sicuro il sistema operativo non può impedire ai processi di richiedere risorse in modo da causare uno stallo. Dato il concetto di stato sicuro, si possono definire gli algoritmi che permettano di evitare le situazioni di stallo, semplicemente assicurando che il sistema rimanga sempre in uno stato sicuro. Un ***algoritmo con grafo di assegnazione delle risorse*** è applicabile ai soli sistemi di assegnazioni delle risorse per singole istanze. Oltre agli archi di richiesta e di assegnazione, si introduce un nuovo tipo di arco detto *arco di reclamo* (*claim edge*), che indica che un processo può richiedere una risorsa in un qualsiasi momento futuro. Per quanto riguarda l'assegnazione delle risorse con più istanze di ciascun tipo di risorsa si può utilizzare l'***algoritmo del banchiere***. Quando si presenta, un nuovo processo deve dichiarare il numero massimo delle istanze di ciascun tipo di risorsa di cui necessita; questo numero non può superare il numero totale delle risorse del sistema. Quando un utente richiede un gruppo di risorse, si deve stabilire se l'assegnazione di tali risorse lasci il sistema in uno stato sicuro. Se si rispettano tali condizioni le risorse vengono assegnate, altrimenti il processo deve attendere che qualche altro processo ne rilasci un numero sufficiente.

Se un sistema non si avvale di un algoritmo per prevenire o evitare le situazioni di stallo, è possibile che si verifichi effettivamente. In tal caso avremo bisogno di un algoritmo di ***rilevamento delle situazioni di stallo*** (*detection*). Se tutte le risorse hanno una sola istanza si può definire un algoritmo di rilevamento di situazioni di stallo che fa uso di una variante del grafo di assegnazione delle risorse, detto *grafo d'attesa*, ottenuta dal grafo di assegnazione delle risorse togliendo i nodi dei tipi di risorse e componendo gli archi tra i processi. Nel sistema esiste uno stallo se e solo se il grafo d'attesa contiene un ciclo. Per individuare le situazioni di stallo, il sistema deve conservare il grafo d'attesa e invocare periodicamente un algoritmo che cerchi un ciclo all'interno del grafo. L'algoritmo per il rilevamento di un ciclo all'interno di un grafo richiede un numero di operazioni dell'ordine di n^2, dove con n si indica il numero dei vertici del grafo. Tale algoritmo non si può applicare ai sistemi di assegnazione delle risorse con più istanze di ciascun tipo di risorsa. Per questi sistemi si utilizza un algoritmo che si serve di strutture dati variabili nel tempo, simili a quelle utilizzate nell'algoritmo del banchiere (vedere sul libro pag. 286). Per sapere quando è necessario ricorrere all'algoritmo di rilevamento si devono considerare i seguenti fattori: la frequenza presunta con la quale si verifica uno stallo e il numero di processi che sarebbero influenzati da tale stallo. Se le situazioni di stallo sono frequenti, è necessario ricorrere spesso all'algoritmo per il loro rilevamento. Naturalmente, l'uso dell'algoritmo di rilevamento per ogni richiesta aumenta notevolmente il carico nei termini di tempo di calcolo. A questo proposito, l'algoritmo di rilevamento viene invocato, di solito, a intervalli meno frequenti, oppure ogni volta che l'utilizzo della CPU scende sotto un determinato valore, poichè uno stallo può rendere inefficienti le prestazioni del sistema e quindi causare una drastica riduzione dell'utilizzo della CPU.

Una volta rilevata, una situazione di stallo si può affrontare in due modi: informare l'operatore della presenza di uno stallo in modo che possa gestirlo manualmente, lasciare che avvenga il ***ripristino da una situazione di stallo*** automaticamente dal sistema. Uno stallo può essere eliminato in due modi. Il primo metodo prevede la *terminazione dei processi* che può avvenire in due modi: o vengono eliminati tutti i processi in stallo (operazione molto onerosa), o viene eliminato un processo per volta fino all'eliminazione del ciclo di stallo (notevole carico in quanto dopo aver terminato ogni processo si deve impiegare un algoritmo di rilevamento per stabilire se esistono ancora processi in stallo). Il secondo metodo prevede l'esercitazione della prelazione sulle risorse, cioè le risorse si sottraggono in successione ad alcuni processi e si assegnano ad altri finchè si ottiene l'interruzione del ciclo di stallo. Si devono considerare i seguenti problemi: selezione di una vittima, cioè occorre stabilire quali risorse e quali processi si devono sottoporre a prelazione e in che ordine; ristabilimento di un precedente stato sicuro, cioè bisogna stabilire che cosa fare con un processo a cui è stata sottratta una risorsa (deve essere ricondotto a uno stato sicuro); attesa indefinita, cioè occorre garantire che le risorse non siano sottratte sempre allo stesso processo.